Approksimointialgoritmit (kevät 2010)
Harjoitus 6 (15. maaliskuuta)
Ratkaisuja (Jouni Siren)

1. Olkoon $G = (V, E)$ kaaripainoilla varustettu verkko ja c painofunktio. Tarkastellaan k-leikkausta, jossa verkon solmut jaetaan erillisiin osajoukkoihin $S_1, \dots, S_k$. Tavoitteena on maksimoida eri joukkojen välisten kaarten painojen summa.

   Jos kaikki solmut sijoitetaan toisistaan riippumatta satunnaisesti joukkoihin S_i, saavutetaan odotusarvoinen approksimointisuhde $(k-1)/k$. Kunkin kaaren päät sijoitetaan nimittäin todennäköisyydellä $(k-1)/k$ eri osajoukkoihin, joten odotusarvon lineaarisuuden takia leikkauksen painon odotusarvoksi saadaan

   $$\sum_{e \in E} \frac{k-1}{k} c(e) = \frac{k-1}{k} \sum_{e \in E} c(e) \geq \frac{k-1}{k} \cdot \mathsf{OPT}.$$

   Poistamalla satunnaisuus saadaan yleistys tehtävän 2.1 algoritmista. Nimetään verkon solmut $v_1, \dots, v_n$ niiden käsittelyjärjestyksen mukaan. Sanotaan, että solmu v_j omistaa kaaren (j', j), jos $j' < j$. Solmu v_j sijoitetaan siihen joukkoon S_i, joka minimoi joukon sisään jäävien solmun v_j omistamien kaarten yhteispainon. Näin ollen leikkaukseen tulee aina vähintään osuus $(k-1)/k$ solmun v_j omistamien kaarten yhteispainosta. Koska jokaisen kaaren omistaa jokin solmu, saadaan näin ratkaisu, jonka kokonaispaino on vähintään $\frac{k-1}{k} \cdot \mathsf{OPT}$.

2. Olkoon koneiden joukko $M = \{1, \dots, m\}$, töiden joukko $J = \{1, \dots, m^2 - m + 1\}$, $p_{i,1} = m$ kaikilla $i \in M$ ja $p_{i,j} = 1$ kaikilla muilla $i \in M$ ja $j \in J$. Tarkastellaan ratkaisua, jossa $x_{i,1} = 1/m$ kaikilla $i \in M$ ja $x_{i,j} = 1$, kun $(m-1)(i-1)+2 \leq j \leq (m-1)i+1$. Muilla pareilla (i,j) pätee $x_{i,j} = 0$. Osoitetaan, että tarkasteltava ratkaisu x on kantaratkaisu ohjelmaan $LP(m)$ (luentojen sivu 199).

   Merkitään $n = m^2 - m + 1$. Käypä ratkaisu x on kantaratkaisu, jos $|M \times J| = mn$ lineaarisesti riippumatonta rajoitetta pätee yhtäsuuruutena.

   (a) Kaikki n muotoa $\sum_{i \in M} x_{i,j} = 1$ olevat rajoitteet ovat väistämättä yhtäsuuruuksia.
   (b) Kaikki m muotoa $\sum_{j \in J} x_{i,j} p_{i,j} \leq m$ olevat rajoitteet ovat yhtäsuuruuksia, sillä jokaisen koneen kuorma on m.
   (c) Muotoa $x_{i,j} \geq 0$ olevista rajoitteista yhtäsuuruuksia ovat ne parit (i,j) joilla kone i ei suorita työtä j. Näitä pareja on $mn - m - (n-1)$ kappaletta.

   Rajoitteista on yhtäsuuruuksia yhteensä $mn + 1$ kappaletta. Mitkä tahansa mn kappaletta näistä ovat lineaarisesti riippumattomia, joten ratkaisu x on kantaratkaisu.

   Tehtävän voi ratkaista myös suoraan. Oletetaan, että x on aito lineaarikombinaatio kahdesta käyvästä ratkaisusta y ja y'. Tarkastellaan ratkaisua y. Koska $x_{i,j} \in \{0, 1\}$ kaikilla $i \in M$ ja $j > 1$, täytyy näillä i ja j päteä myös $y_{i,j} = x_{i,j}$. Jäljellä ovat enää muuttujat $y_{i,1}$ kaikilla $i \in M$.

   Tarkastellaan rajoitetta (b) jollain $i' \in M$. Nyt

   $$\sum_{j \in J} y_{i',j} p_{i',j} = \sum_{j > 1} x_{i',j} p_{i',j} + y_{i',1} p_{i',1} = m - 1 + y_{i',1} \cdot m \leq m,$$

   joten täytyy olla $y_{i',1} \leq 1/m$. Toisaalta rajoite (a) arvolla $j = 1$ tarkoittaa $\sum_{i \in M} y_{i,1} = 1$, mistä seuraa $i' = 1/m$. Näin ollen $y = x$ ja vastaavasti myös $y' = x$. Ratkaisu x ei siis ole aito lineaarikombinaatio ratkaisuista y ja y' vaan kantaratkaisu.

3. Muodostetaan approksimointisuhteen säilyttävä palautus solmupeiteongelmasta puiden monileikkausongelmaan.

   Olkoon $G = (V, E)$ solmupeiteongelman verkko ja cost solmujen painofunktio. Muodostetaan puu T asettamalla uusi solmu u sen juureksi ja joukon V solmut juuren lapsiksi. Asetetaan $\mathsf{cost}(u, v) = \mathsf{cost}(v)$ kaikilla $v \in V$. Otetaan puun T monileikkausongelman päätesolmupareiksi jokainen kaari $(v, v') \in E$. Nyt $V' \subseteq V$ on solmupeiteongelman ratkaisu jos ja vain jos kaaret (u, v), missä $v \in V'$, muodostavat monileikkausongelman ratkaisun. Lisäksi molemmat ratkaisut ovat saman hintaisia, joten kysymys on approksimointisuhteen säilyttävästä palautuksesta solmupeiteongelmasta monileikkausongelmaan.

Jos puun korkeus on 1, toimii yllä oleva palautus myös toiseen suuntaan. Yleisissä puissa vastaava suoraviivainen palautus ei onnistu. Sen sijaan puiden monileikkausongelma voidaan palauttaa joukkopeiteongelmaan valitsemalla perusjoukon alkioiksi päätesolmuparit ja osajoukoiksi puun kaaret. Kukin osajoukko sisältää ne päätesolmuparit, joiden väliseen polkuun kaari kuuluu. Osajoukon hinnaksi asetetaan sitä vastaavan kaaren paino.

4. Osoitetaan, että käänteisessä lisäysjärjestyksessä tapahtuva ylimääräisten kaarten poisto on välttämätön luentojen sivun 216 algoritmin approksimointitakuulle.

   Oletetaan ensin, että poistoa ei tehdä lainkaan. Tarkastellaan puuta, joka sisältää mielivaltaisen pitkän polun solmuparin (s, t) välillä. Asetetaan jokaisen polulla olevan kaaren kapasiteetiksi 1. Algoritmi valitsee vaiheessa 2 kaikki polulla olevat kaaret mukaan leikkaukseen, koska solmujen s ja t välinen vuo kyllästää ne samanaikaisesti. Kuitenkin jo yhden kaaren valitseminen riittäisi erottamaan solmut toisistaan.

   Oletetaan sitten, että poisto tehdään lisäysjärjestyksessä. Valitaan puuhun solmut u_i, v_i ja w_i kaikilla $1 \leq i \leq n$. Asetetaan $c(u_i, v_i) = c(v_i, w_i) = 1$ kaikilla $1 \leq i \leq n$ ja $c(v_i, v_{i+1}) = n$ kaikilla $1 \leq i < n$. Valitaan puun juureksi v_n. Toisistaan erotettavat parit ovat $p_i = (u_i, w_{i+1})$ kaikilla $1 \leq i < n$ sekä (v_1, v_n).

   Parit p_i käsitellään nousevassa järjestyksessä solmun v_{i+1} kohdalla. Niiden välillä saadaan kuljetettua yhden yksikön verran vuota, mikä kyllästää kaaret (u_i, v_i) sekä (v_{i+1}, w_{i+1}). Viimeiseksi käsitellään solmun v_n yhteydessä pari (v_1, v_n), mikä kyllästää kaaret (v_i, v_{i+1}) kaikilla $1 \leq i < n$ viemällä niiden läpi $n - 1$ yksikköä vuota.

   Poistovaiheessa kaaret (u_i, v_i) ja (v_{i+1}, w_{i+1}) poistetaan leikkauksesta kaikilla $1 \leq i < n$, sillä kaari (v_i, v_{i+1}) erottaa vielä solmuparin p_i toisistaan. Sen sijaan mitään kaarista (v_i, v_{i+1}) ei voida poistaa, sillä (v_i, v_{i+1}) on ainoa leikkauksessa mukana oleva kaari parin p_i välisellä polulla kaikilla $1 \leq i < n$. Ratkaisuna saadaan siis leikkaus, jonka kapasiteetti on $n(n - 1)$. Toisaalta valitsemalla vain kaari (v_1, v_2) sekä kaaret (u_i, v_i) kaikilla $1 < i < n$ saataisiin leikkaus, jonka kapasiteetti on $2n - 2$, joten algoritmin löytämä ratkaisu on $n/2$ kertaa huonompi.